An Analysis of Secured Routing Technique in Mobile Ad Hoc Networks

Gowsiga Subramaniam*[1], SenthilKumar Ponnusamy[2], Muneeswari A[3]

[1]Information Technology, Saveetha Engineering College, Chennai, Tamilnadu, India

gowsiga@gmail.com

[2]Information Technology, SKR Engineering College, Chennai, Tamilnadu, India

drsenthilkumar2010@gmail.com

[3] Saveetha School of Law,Saveetha University, Tamil Nadu, India

Abstract

A mobile ad hoc network (MANET) is a collection of wireless mobile nodes which have the ability to communicate with each other without having fixed network infrastructure or any central base station. Individual nodes are responsible for dynamically discovering other nodes that they can directly communicate with. Communication between nodes may be achieved by means of Route Discovery and maintenance. Because of mobility nature, discovery task is more complex. A black hole is a malicious node that falsely replies to any route requests without having active route to specified destination and drops all the receiving packets. Black Hole Attack can be easily deployed by the adversary. If these malicious nodes work together as a group then the damage will be very serious. This type of attack is called cooperative black hole attack. A neighborhood based method to detect whether there exists a black hole attack and a routing recovery protocol to set up a correct path to the true destination. We propose a complete protocol to detect a chain of cooperating malicious nodes in an ad hoc network that disrupts transmission of data by feeding wrong routing information. In this model, the nodes authenticate each other by issuing security certificate with digital signature to all the other nodes in the network. This model is capable of detecting and removing black hole nodes in the MANET also with a secured route discovery by means of an algorithm termed as endairA algorithm.

Keywords

EndairA; Digital Signatures; Black Hole; DRI

Introduction

Mobile hosts and wireless networking hardware are becoming widely available, and extensive work has been done recently in integrating these elements into traditional networks such as the Internet. Oftentimes, however, mobile users will want to communicate in situations in which no fixed wired infrastructure such as this is available, either because it may not be economically practical or physically possible to provide the necessary infrastructure or because the expediency of the situation does not permit its installation.

For example, a class of students may need to interact when lecture, friends or business associates may run into each other in an airport terminal and wish to share files, or a group of emergency rescue workers may need to be quickly deployed after an earthquake or flood. In such situations, a collection of mobile hosts with wireless network interfaces may form a temporary network without the aid of any established infrastructure or centralized administration. This type of wireless network is known as an ad hoc network.

Efficient communication through networking has become the order of the day. Networking, till a few decades back was confined to the wires that imposed a limitation on the positioning of users. With the advancement of technology in strides, networking has evolved into the wireless mode, wherein it is not required for the users to be stationary. An ad hoc network is a group of wireless mobile computers (or nodes); in which nodes cooperate by forwarding packets for each other to allow them to communicate beyond direct wireless transmission range.

The main aim is to concentrate on a design of a more secure routing protocol for ad hoc networks. The design of routing protocol is based on On-demand source routing. The function of the routing protocol in ad hoc network is to establish route between nodes. Several 'secure' routing protocols have been proposed for ad hoc networks such as AODV, DSR, ZRP, TORA, DSDV, TBRPF, Ariadne and others. These secure routing protocols still have security vulnerabilities, and can be attacked. In this paper we implement a more secure routing protocol for ad hoc networks. The protocol is implemented using GloMoSim network simulator.

Routing Algorithm

Route discovery can be proactive or reactive (on-demand). Proactive routing is usually table driven with reactive algorithms; in which routes are discovered only when needed. Proactive routing protocols maintain routes to all destinations, regardless of whether or not these routes are needed. In order to maintain correct route information, a node must periodically send control messages. Therefore, proactive routing protocols may waste bandwidth since control messages are sent out unnecessarily when there is no data traffic. The main advantage of this category of protocols is that hosts can quickly obtain route information and quickly establish a session. Reactive routing protocols can dramatically reduce routing overhead because they do not need to search for and maintain the routes on which there is no data traffic. This property is very appealing in the resource-limited environment.

Routing is a basic network functionality that supports communication. In MANETs, each node acts as a router forwarding data to other nodes.

It distinguishes three basic phases in routing:

1) Route discovery in which one or more routes (of adjacent nodes) that link a source *S* to a target *T* are sought,

2) Route maintenance in which broken links of established routes are fixed, and

3) Packet forwarding in which communication is achieved via established routes.

Route discovery can be proactive or reactive (on-demand). Proactive routing is usually table driven with reactive algorithms; routes are discovered only when needed.

The Source Routing Protocol (SRP)

The Source Routing Protocol algorithm is an on-demand algorithm, which enables dynamic, self-starting, multihop routing to be established when a source sensor node wishes to send a data packet. All the routing messages in SRP are small and have fixed length.In this way less transmission energy is needed for the routing overhead, which is especially important at the source initiated request flooding. The SRP algorithm has three phrases: Route Setup, Route Maintenance and Route Re-establishment.

In SRP, route requests generated by a source *S* are protected by Message Authentication Codes (MACs) computed using a key shared with the target *T.* Requests are broadcast to all the neighbors of *S.* Each neighbor that receives a request for the first time appends its identifier to the request and rebroadcasts it. Intermediate nodes do the same. The MAC in the request is not checked because only *S* and *T* know the key used to compute it. When this request reaches the target *T*, its MAC is checked by *T*. If it is valid, then it is assumed by the target that all adjacent pairs of nodes on the path of the route request are neighbors. Such paths are called valid or plausible routes. The target *T* replaces the MAC of a valid route request with an MAC computed with the same key that authenticates the route. This is then sent back (upstream) to *S* using the reverse route. For example, a route request that reaches an intermediate node *Xj* is of the form

$$\mathbf{msg_{S,T,rreq}=(rreq,S,T,id,sn,X_1,....X_j,mac_S)}$$

with *id* a randomly generated route identifier, *sn* a session number, and macS an MAC on (rreq, S,T, id, sn) computed by *S* using a key shared with *T*. If *S,X1, . . .,Xp; T* is a discovered route, then the route reply of the target *T* has the following fixed form for all intermediate nodes Xj, $1 \le j \le p$:

$$\mathbf{msg_{S,T,rrep}=(rrep,S,T,id,sn,X_1,...Xp,mac_T)}$$

where mac_T is an MAC computed by *T* with the key shared with *S* on the message fields preceding it.

Ariadne

Ariadne is an on-demand routing algorithm based on the Dynamic Source Routing (DSR) protocol. There are several variants of Ariadne, depending on which mode of authentication is used to protect route requests: one uses digital signatures, one TESLA, and one uses MACs. The MAC version has an optimized variant that uses iterated MAC computations instead of several independent MACs. In addition to being more efficient, the iterated MAC version has superior security characteristics when compared to the no optimized version. A typical route request that reaches an intermediate node Xj, $1 \le j \le p$, on the route $S = X_0, X_1, \ldots, Xp$, $Xp_{+1} = T$ is of the form

$$\mathbf{msg_{S,\,T,rreq}=(rreq,S,T,id,X_1,..Xj,\ macsxi..._{Xj})}$$

where $mac_{SX1...Xj}$ is the MAC computed by *Xj* with a key it shares with *T* on the route request received from X_{j-1}.The route reply of *T* is

$$\mathbf{msg_{S,\,T,rrep}=(rrep,S,T,id,X_1,..Xp,mac_T)}$$

where mac_T is an MAC computed by T with a key shared with S on the message field that precedes it (rrep, S, T, id,X1, . . .,Xp).

EndairA

EndairA is one of the most secure on-demand ad hoc network source routing protocols which provides several defense mechanisms against so many types of attacks. In this paper, we prove the vulnerability of endairA to the tunneling attack by presenting an attack scenario against it. We also propose a new security mechanism to defend it against the tunneling attack by the utilization of the delay between receiving and sending its control packets computed locally by all intermediate nodes. Our proposed security mechanism can detect probable tunnels in the route as well as approximate locations of the adversarial nodes. It needs no time synchronization between mobile nodes of the network. It also does not change the number of control packets involved in endairA and only modifies the RREP messages slightly. Fundamentally, endairA (and the ABV model) was developed to deal with a class of hidden channels, the intrinsic hidden channels of a wireless broadcast medium in a neighborhood. However, security is not achieved because other hidden channels remain present.

Dtection of Back Hle Atack

On receiving the monitor message neighbors of the source node checks whether it is the neighbor of the next hop node in route or not. If it is neighbor of the hop node in route then it starts monitoring the action of the node. It at first initializes a counter to count the no. of the data packets forwarded by the node also infers the id of the next node to which it is forwarding the data. To monitor nodes can maintain a copy of the neighbor's routing table and determine the next-hop node to which the neighbor should forward the packet; if the packet is not overheard as being forwarded, it is considered to have been dropped. Also the monitor nodes again broadcast a monitor message to all its neighbors containing the id of the next node to which this node is forwarding the data, instructing them to monitor the action of the next node. This process will continue until the next node is the destination node. If the receiving node of the monitor message is not the neighbor of the next hop node in route, it simply forwards the message to all its neighbors.

Whenever a source node wishes to initiate the gray/black hole detection and removal process, it broadcasts a query message to all its neighbors and sets a time out for the receipt of the result message from the monitoring nodes. When the timeout didn't expire each time, a result message or the node is malicious message which received for any node source node will append that node in its findMalicious Table and initialize the voteCount as 1 if it is not already there, otherwise increments its voteCount by 1 and check if voteCount is greater than a predefined thresholdCount or not. If greater, then source node will remove that node from the findMalicious table and enter it into the Black/Gray Hole table. Broadcasts that the node is malicious to the network and modify the malicious status of that route by setting the findHoleStatus as true for that route in its routing table. When the timeout expired source node will start voting for the nodes left in the findMalicious table. It broadcasts vote request message to the network containing the id of each node in the findMalicious table one by one.Setting a timeout for the receipt of the vote reply and on receiving a reply voteCount is incremented by 1.

Check if the voteCount is greater than a predefined thresholdCount removing that node from the findMalicious table and entering it into the Black/Gray Hole table.It also broadcasts that the node is malicious to the network and the malicious status of that route by setting the findHoleStatus as true for that route in its routing table.

Finally the source node checks the findHoleStatus of theroute and if it is true then it terminates sending data until it finds a new route to the destination. If it is not true then it retries sending data of the same block.

On receiving a vote request for any node, a regular node in the network checks their Black/Gray Hole table. If an entry for that node is found, it response to the source node (i.e. the generator of the vote request) via a vote reply message. Here we assume that if the node is not a newly joined node then there is a possibility that node has traversed from the different region of the network. So any other node in the network may have used this node for forwarding traffic and found it as malicious.

On receiving a node which is malicious message, all regular nodes in the network first check if they already have an entry for the node in their Black/Gray Hole table. If not then they make an entry for that node in their findMalicious table and initialize voteCount as 1. If the node already exists in any of the above tables ignoring the message. We are doing so, because if we block list the node or increment its voteCount then there is a chance of completely banning a legitimate node from the network by false probing.

Here in our method we propose to modify AODV protocol by introducing three more tables maintained at

each node. Table 1 is DRI (Data Routing Information) table maintained at each node for the purpose of monitoring each of its neighbors. Table4 is the findMalicious table maintained at s which keeps the track of the nodes suspected as malicious with their voteCount.

And the Black/Gray hole table which keeps the track of the black listed nodes. We also modified the routing table of the AODV by adding a new field called findHoleStatus which is set as true if malicious nodes found in the route. With the help of the following Fig.1 which shows a current network topology, each of the tables is depicted below.

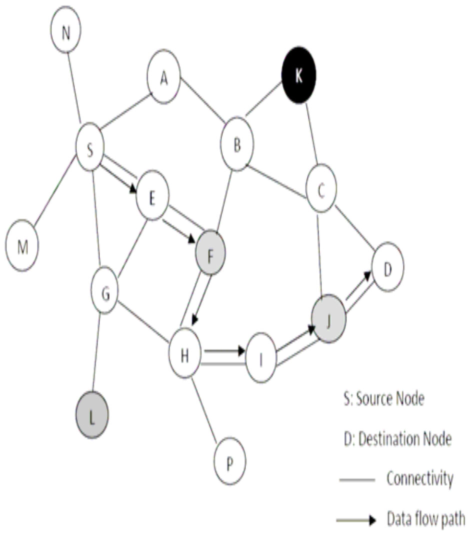


FIG. 1NETWORK TOPOLOGY

TABLE 1 DATA ROUTING TABLE FOR S

Destination Node ID	Route	findHoleStatus
D	E,F,H,I,J	False
P	A,B,F,H	False
J	G,H	False

TABLE 2 LIST OF NEIGHBORS MAINTAINED AT S

NEIGHBOUR NODE ID
E
G
A
M
N

TABLE 3 DATA ROUTING INFORMATION TABLE MAINTAINED AT NODE B FOR MONITORING NEIGHBORS

Monitored Node ID	Next Node ID	Data Count
F	H	5
K	NULL	0

TABLE 4 FIND MALICIOUS TABLE MAINTAINED AT S

Node ID	DataCount
F	2
J	1

TABLE 5 BLACK/GRAY HOLE TABLE MAINTAINED AT S

Node ID
L
K

Algorithm for Detecting Gray/Black Hole

Action by Source Node *S*

Step 1: Divides the data packets to be sent in ***k*** equal parts.

DATA [1,….,K];

Initialize **i = 1;**

Comment: Chose window size *w*, If total no of data packets **n** then ***k*** **= ceiling (n/w)**

Step 2: Send ***start(S,D,ni)*** message to the destination node ***D***. Where ***ni*** is the no of data packets to be sent in current block.

Step 3: Broadcast ***monitor (S, D, NNR)*** message to all its neighbors. Instructing neighbors to monitor next node in the route (**NNR**).

Step 4: Starts transmitting data packets from the block

Data[i] to ***D***.

Step 5: Sets timeout **TS** for the receipt of the ***end (D, S,d_count)*** message containing ***d_count***, no of data packets received by D.

Step 6: If **TS** not expired and ***end*** message received,

if $(ni\,(1-\mu) \leq d\_count)$

Increment ***i*** by **1** and go to **Step 8**.

else Start Gray/Black hole removal process.

Comment: Where μ is a threshold value ranging between 0 and 1 which indicates the fraction of total packets gets lost due to error prone wireless channel. If we assume that μ is the

permissible packet loss in each node in the route then$\mu = 1-(1-\mu)N$, where N is the total no of nodes in the route (hop count).

Step 7: If **TS** expired and ***end*** message not received then start Gray/Black hole removal process.

Step 8: Continues from **Step 2** when ***i*** less than equal to ***k.***

Step 9: Terminates *S*'s action.

<u>Action by Destination Node D</u>

Step 1 On receiving ***start(S,D,ni)*** message from *S*

extracts ***ni***

Initialize **d_count = 0.**

Step 2: Sets timeout **TD** for the receipt of the current data

sample and waits for the data packets.

Step 3: When **TD** not expired and a data packet received

Update **d_count += 1**

Step 4: When **TD** expired send ***end(D, S, d_count)*** message to **S.**

Step 9: Terminates **D**'s action.

<u>Action by neighbors On receiving *monitor (S, D, NNR)* message</u>

Step 1 On receiving ***monitor (S, D, NNR)*** message nodes

extracts the id of the next node in the route **NNR**, source

node id **S** and destination node id **D**.

Step 2: If the receiving node is neighbor of **NNR** then,

Step 2.1: Turn on Promiscus mode**.**

Step 2.2: Initialize **dataCountNNR = 0**.

Step 2.3: Find next node id **Nnext** to which **NNR** is forwarding the data packets.

Step 2.4: start counting data packets by incrementing

dataCountNNR += 1.

Step 2.5.: If **Nnext** is not destination node **D** then

Step 2.5.1: Broadcast ***monitor (S, D, NNR)*** message to

all its neighbors replacing **NNR** by **Nnext**.

Step 3: Else Rebroadcast ***monitor (S, D, NNR)*** message to

all its neighbors.

Step 4: Terminates its action.

<u>Gray/Black Hole Removal process</u>

Action by Source Node S

Step 1: Broadcast ***query(S, D, NRREP, ni)*** message to all its

neighbors. Where **NRREP** is the id of the node sending route reply message to ***S***.

Step 2: Sets timeout **TRES** for the receipt of the ***result (MN, S,NRREP)*** message from the monitoring node **MN.**

Step 3: When **TRES** not expired and ***result*** message received or ***"NRREP Malicious"*** received then extracts **NRREP**.

Step 3.1 If **NRREP** already exists in **FindMalicious** table

Step 3.1.1: Then increment **voteCount** for **NRREP** by 1.

Step 3.1.2: If **votecount >= thresholdCount**

Step 3.1.2.1: Remove **NRREP** from **FindMalicious** table and append **NRREP** in **Gray/BlackHole** table.

Step 3.1.2.2: Broadcast ***"NRREP Malicious"*** to the Network.

Step 3.1.2.3: Set **findHoleStatus = true** in the routing table of S for the route to **D.**

Step 3.2: Else

Step3.2.1: Append **NRREP** in **FindMalicious.**

Step 3.2.2: Initialize **voteCount =** 1.

Step 4: Initialize **j = 1**.

Step 5: When **j <=** length of **FindMalicious** table

Step 5.1: Broadcast ***VREQ(S, Nj)*** to the network requesting other nodes in the network to vote for **Nj** if it is malicious.

Step 5.2: Sets timeout **TVREP** for reply from the network ***VREP(RN, S, Nj)*** where **RN** is id of any regular node in the network.

Step 5.3: When **TVREP** not expired and ***VREP*** message received then

Step 5.3.1: increment **voteCount** for **Nj** by **1.**

Step 5.4: If **voteCount >= thresholdCount**

Step 5.4.1: Remove **NRREP** from **FindMalicious** table and append **NRREP** in **Gray/BlackHole** table.

Step 5.4.2: Broadcast ***"NRREP Malicious"*** to the Network.

Step 5.4.3: Set **findHoleStatus = true** in the routing table of S for the route to **D.**

Step 5.5: Increment **j** by **1.**

Step 6: If **findHoleStatus** is **True**

Step 6.1: Terminate sending data. Find new route to **D**.

Step 7: Resume its normal action.

Action by Neighbors on receiving on receiving *query(S, D,NRREP, ni)*message

Step 1: On receiving ***query(S, D, NRREP, ni)*** message nodes

extracts **NRREP** (id of the node sending route reply message to **D**), **S**, **D** and **ni**(no of data packets sent to D).

Step 2: If the receiving node is neighbor of **NRREP** then,

Step 2.1: If $ni\ (1-\mu\ ) \le dataCount$

Step 2.1.1: when **Nnext** is not **D**

Step 2.1.1.1: Broadcast ***query(S, D, NRREP, ni)*** message to all its neighbors replacing **NRREP** by **Nnext**.

Step 2.2: Else

Step 2.2.1: If **Nnext** equals to **NULL** then **Nnext** itself droping all the packects

Step 2.2.1.1: Reply ***"NRREP Malicious"*** to S.

Step 2.2.2: Else

Step 2.2.2.1: Reply ***result (MN, S, NRREP)*** to **S,** which means **NRREP** may be malicious.

Step 2.2.2.2: Broadcast ***query(S, D, NRREP, ni)*** message to all its neighbors replacing **NRREP** by **Nnext** and **ni** by **dataCount** for **NRREP.**

Step 3: If the receiving node is not neighbor of **NRREP** then

broadcast ***query(S, D, NRREP, ni)*** message to all its neighbors.

Step 4: Terminates its action.

Action by any regular nodes (RN) on receiving on receiving

VREQ(S, Nj) message

Step 1 On receiving ***VREQ(S, Nj)*** message nodes extracts **Nj**

Step 2: If **Nj** exists in **Gray/BlackHole** table

Step2.1: Reply ***VREP(RN, S, Nj)*** to **S**.

Step 3: Terminates its action.

Action by any regular nodes (RN) on receiving on receiving

"NRREP Malicious"

Step 1 On receiving **"NRREP Malicious"** all regular nodes in the network check **Gray/BlackHole** table**.**

Step 2: If **NRREP** not exists in **Gray/BlackHole** table, then

Step 2.1: If **NRREP** not exists in **FindMalicious** table.

Step 2.1.1: Append **NRREP** in **FindMalicious** table.

Step 2.2.2: Initialize **voteCount = 1**.

Step 3: Terminates its action.

Analysis of Endaira

This implies that the route can be uniquely partitioned as follows: each partition consists of a single on compromised identifier (label) or a sequence of consecutive compromised identifiers. A plausible route is one whose partitions correspond to that of a real route that physically exists in the network. The security statement of endairA is that it only accepts plausible routes. Note that this statement also fails to consider an adversarial lengthening of a route by assignment of multiple labels to a single compromised network node as an attack. Again, this is a strong restriction on the security guarantees that the ABV model can provide, but we also follow this paradigm because we wish to show that endairA fails in the exact model in [15].

An Attack on EndairA

This is a hidden channel attack that does not require out-of-band resources. Consider an instance of endairA with source node *S* and let *(S; A; X; B; Y; D; T)* be a sequence of identifiers of pair wise neighbor nodes in which only *X; Y* are faulty. In the attack, when the second faulty node *Y* receives

$$\mathbf{msg_{S,T,rreq}=(rreq,S,T,id,A,X,B)}$$

it drops node *B* from the listing and transmits

$$\mathbf{msg_{S,T,rrep}=(rrep,S,T,id,A,X,Y,D,sig_T,\ sig_D\ )}$$

Eventually, the route request will reach the target T, which will compute and send back a route reply. Node *Y* will then receive from *D*

$$\mathbf{msg}_{D,A,,rreq}\mathbf{=(rreq,D,A,id)}$$

Now, *Y* can obviously attach its label and signature to this reply and transmit to *B* the extended reply, but *B* will not retransmit it because *B* is not included in the listing. Eventually, *X* will be able to reconstruct these signatures and can then generate the route reply

$$\mathbf{msg}_{S,T,rrep}\mathbf{=(rrep,S,T,id,A,X,Y,D,sig}_{T}\mathbf{,\ sig}_{D}\mathbf{,\ sig}_{Y}\mathbf{,\ sig}_{X}\mathbf{)}$$

this is sent back to the source *S* and validated.

Hidden Channel and Concurrency Attacks

In all the attacks described above, including the attacks adversarial nodes succeed in shortening plausible routes by removing intermediate nodes. The adversarial nodes use hidden channels to communicate and transfer the necessary data (signatures, etc.). The hidden channels that we considered above do not use out-of-band resources; although this is an obvious alternative. Let us now pursue our earlier discussion on interleaving protocol instances. In a networking environment, one should expect that several instantiations of a routing protocol are executed. Some may involve route discovery, while others route maintenance, data communication, or general network applications. It makes no sense to require that route communication can only start when all the other route discovery instantiations (and network applications) have been completed.

Secure Route Discovery Challenges

The routing must be resilient to threats and vulnerabilities. It must have inbuilt capability to avoid resource consumption, denial of service, impersonation and similar attacks possible against a network. Our argument about the impossibility of secure discovery of routes is simple and has been articulated throughout the paper. We base it on the fact that every route discovery algorithm is, in practice, vulnerable to attacks that exploit alternative communication channels to articulate distributed attacks by “encapsulating” and tunneling routing requests. Therefore, it does not seem possible to capture or “model out” Sybil and wormhole attacks from pure-protocol-based security models. The purpose of routing is to establish a communication infrastructure, and it is always reasonable to assume the existence of alternative communication channels, namely those that route discovery will establish. Even though it is not possible to discover secure routes in general MANETs, there are several other approaches that could be used to establish secure communication channels.

Proposed Result and Comparative Study

At the end of the result, we expect to have a network which provides more security for route discovery in Mobile Ad hoc network.

Proposed Result

A new security framework was tailored for on-demand route discovery protocols in MANETs. This represents the first effort toward a formal security model that can deal with concurrent attacks and is successful in mitigating a class of hidden channel attacks—the attacks that are intrinsic to the wireless broadcast medium in a neighborhood.

SRP is simple but it does not prevent the manipulation of mutable information added by intermediate nodes. The message authentication code is used for the authentication. The system was used to authenticate the route by Message authentication codes (MACs) and Digital Signatures.

The Blowfish algorithm is used to provide confidentiality for the routing packets. Authentication and confidentiality operations are used to protect active attacks on route discovery. The passive attacks are handled by the neighbor verification method.

Comparative Study

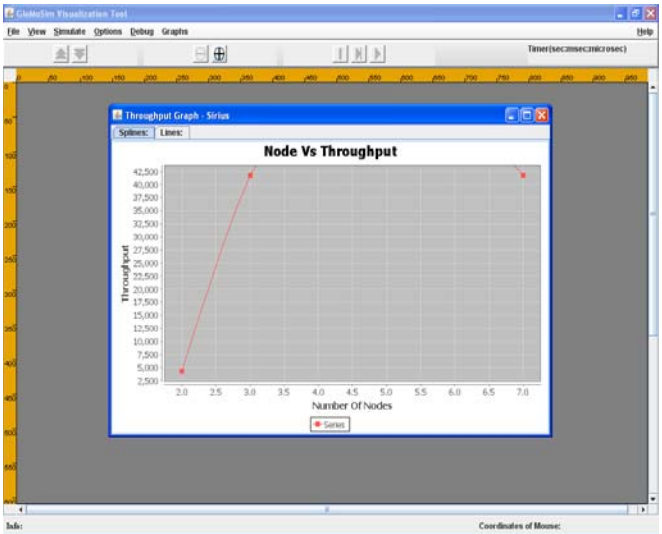
TABLE 6 COMPARISON OF VARIOUS PROTOCOLS

S. No.	ALGORITHM/ PROTOCOL	Accuracy	Speed	Cost
1.	Secure Routing Protocol	Med*	High	High
2.	Ariadne Routing Protocol	Med*	Low	High
3.	Per-hop hash mechanism	Med*	Low	Med*
4.	Symmetric-key broadcast authentication with TESLA	High	Low	Med*
5.	endairA Routing Protocol	High	High	Med*
6.	Symmetric Block Cipher	High	High	Med*
7.	Cryptographic hash function	Low	Med*	Low

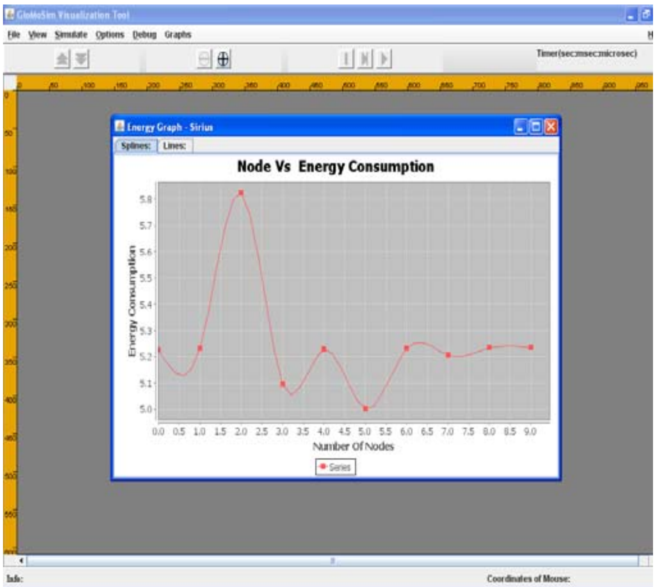
*Med - Medium

Conclusion

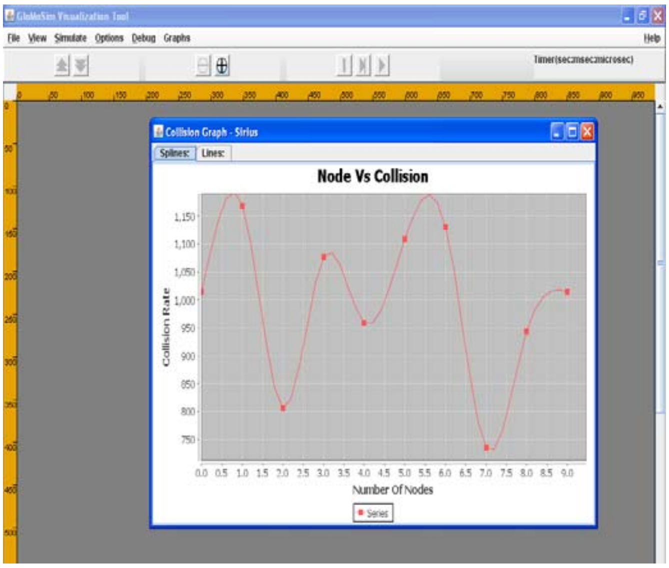
A new security framework was tailored for on-demand route discovery protocols in MANETs. This is successful in mitigating a class of hidden channel attacks—the attacks that are intrinsic to the wireless broadcast medium in a neighborhood. In the proposed formal model, it is impossible to prevent adversarial nodes from breaking up routes by inserting non existing links. To address this shortcoming, either more flexible definitions of routes must be employed or it becomes necessary to address global threats directly, such as those posed by Sybil, wormhole, and more generally, man-in-the-middle attacks. Following graph represents the various activities of different protocols for nodes in the network. Here GlomoSim is used to analysis the results for Throughput (a), Energy consumption (b) and collision(c) shown in Fig.2.



(a) THROUGHPUT



(b) ENERGY CONSUMPTION



(c) COLLISION GRAPH

FIG.2 GRAPH REPRESENTATION

ACKNOWLEDGEMENT

The author wish to thank the anonymous reviewers for their useful suggestions that helped in improving the quality of this paper. The author would like to express her sincere appreciation to her advisor, Professor Dr. P.Senthilkumar, who guided me through this research, inspired and motivated me. Last but not least, the researchers would like to thank the management, the principal and Head of the Department of Saveetha Engineering College, Chennai for supporting this research.

REFERENCES

Acs, G., Buttya´n.L. and Vajda, I. (2005) ‘Provable Security of On- Demand Distance Vector Routing in Wireless Ad Hoc Networks’,Proc. European Workshop Security and Privacy in Ad Hoc and Sensor Networks (ESAS ’05), pp. 113-127.

Acs, G., Buttya´n.L. and Vajda, I. (2006) ‘Modeling Adversaries and Security Objectives for Routing Protocols in Wireless Sensor Networks’, Proc. Workshop Security in Ad Hoc and Sensor Networks (SASN ’06).

Beaver, D. and Haber, S. (1992) ‘Cryptographic Protocols Provably Secure against Dynamic Adversaries’, Proc. Conf. Advances in Cryptology (EUROCRYPT ’92), pp. 307-323.

Burmester, M., van Le, T., and Weir, M. (2003). ‘Tracing Byzantine Faults in Ad Hoc Networks’, Proc. Conf. Computer, Network and Information Security 2003, pp. 43-46.

Burmester, M., van Le, T., and Yasinsac, A. (2007) 'Adaptive Gossip Protocols: Managing Security and Redundancy in Dense Ad Hoc Networks', J. Ad Hoc Networks, vol. 5, no. 3, pp. 286-297.

Buttya´n.L. and Vajda, I. (2004) 'Towards Provable Security for Ad Hoc Routing Protocols', Proc. ACM Workshop Ad Hoc and Sensor Networks (SASN '04).

Douceur.J.R. (2002) 'The Sybil Attack', Proc. First International Workshop Peer-to- Peer Systems (IPTPS '02), pp. 252-260.

Galuba.W,P.Papadimitratos,M. Poturalski ,K.Aberer (2010) 'Castor: Scalable Secure Routing for Ad Hoc Networks',pp.1-9.

Hall,J., Barbeau,M., and Kranakis,E. (2004) 'Enhancing Intrusion Detection in Wireless Networks Using Radio Frequency Fingerprinting', Proc. IASTED Int'l Conf. Comm., Internet, and Information Technology.

Hu.Y.C., Johnson, D.B., and Perrig, A. (2003) 'SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks', Ad Hoc Networks, vol. 1, no. 1, pp. 175-192.

Hu.Y.C., Perrig, A., and Johnson, D. (2002) 'Ariadne: A Secure on- Demand Routing Protocol for Ad Hoc Networks', Proc. ACM MobiCom.

Hu.Y.C., Perrig, A., and Johnson, D. (2003) 'Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks', Proc. IEEE INFOCOM, pp. 1976-1986.

Johnson, D. and Maltz D. (1996) 'Dynamic Source Routing in Ad Hoc Wireless Networks', Mobile Computing, Kluwer Academic Publishers.

Jong-Moon Chung, Dong-Chul Go (2012)' Stochastic Vector Mobility Model for Mobile and Vehicular Ad Hoc Network Simulation', IEEE Transactions on Mobile Computing, vol. 11, no. 10, pp. 1494 - 1507 .

Kaynia, M. Jindal, N. ; Oien, G.E. (2011) ' Improving the Performance of Wireless Ad Hoc Networks Through MAC Layer Design ' IEEE Transactions on Wireless networks, vol. 10, no. 1, pp. 240 - 252 .

Khalil, I., Bagchi, S., (2011) 'Stealthy Attacks in Wireless Ad Hoc Networks: Detection and Counter measure ' IEEE Transactions on Mobile Computing, vol. 10, no. 8, pp. 1096 - 1112 .

Mike Burmester, Breno de Medeiros (2009) 'On the Security of Route Discovery in MANETs', IEEE Transactions on Mobile Computing, vol. 8, no. 9, pp. 1180-1188.

Papadimitratos, P. and Haas, Z. (2002) 'Secure Routing for Mobile Ad Hoc Networks', Proc. SCS Comm. Networks and Distributed Systems Modeling and Simulation Conf. (CNDS '02).

Perkins C.E. and Bhagwat, P. (1994), 'Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers', Proc. ACM SIGCOMM,

Perkins, C. (1997) 'Ad-Hoc On-Demand Distance Vector Routing', Proc. Military Comm. Conf. (MILCOM '97), panel on ad hoc networks.

Perrig, J.T.A., Canetti, R., and Song, D. (2000) 'Efficient Authentication and Signing of Multicast Streams over Lossy Channels', Proc. IEEE Symp. Security and Privacy, pp. 56-73.

Pfitzmann, B. and Waidner, M. (2000) 'Composition and Integrity Preservation of Secure Reactive Systems', Proc. ACM Conf. Computer and Comm. Security, pp. 245-254.

Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C., and Belding-Royer.E.M. (2002) 'A Secure Routing Protocol for Ad Hoc Networks', Proc. IEEE Int'l Conf. Network Protocols (ICNP '02).

Xiang, Xiaojing, Wang, Xin , Yang, Yuanyuan (2010) 'Stateless Multicasting in Mobile Ad Hoc Networks' IEEE Transactions on Computers, vol. 59, no. 8, pp. 1076 – 1090.

Younis, M., Farrag, O. , Althouse, B. (2012) 'TAM: A Tiered Authentication of Multicast Protocol for Ad-Hoc Networks' IEEE Transactions on Network and Service Management, vol. 9, no. 1, pp. 100 - 113 .

Gowsiga Subramaniam received the BTech degree in Information Technology in May 2006 and Masters Degree in Computer Science and Engineering in May 2010 from Velalar College of Engineering and Technology, Erode, Tamilnadu, India. She is currently pursuing the PhD degree in Computer Science and Engineering at Anna University, Chennai. She was a lecturer at Kumaraguru College of Technology, Coimbatore during August 2006 – May 2008. She is working as Assistant Professor at Saveetha Engineering College from July 2010 to till now. Her area of research is in mobile computing with primary focus on security in mobile ad hoc networks.

SenthilKumar Ponnusamy is currently working as Professor and Head of the Department of Information Technology, SKR Engineering College, Chennai. He received his ME in 2002 from Arulmigu Kalasalingam college of engineering, Krishnan kovil and PhD in 2010 from Bharathiyar University, India. He has 13 years of experience in various engineering college. He has published 26 papers in various national and International journal and conferences. His area of research is network.

Muneeswari Senthil Kumar is currently working as Assistant Professor, Saveetha University Chennai. She received her BE in 2003 from Arulmigu Kalasalingam college of engineering, Krishnan kovil and MBA in 2011 from Velammal college of engineering & Technology, India. She has 10 years of experience in various engineering college. Her area of research is network.